

ウェーブレット画像変換の学習ソフトの試作

加治佐 清光[†]

Implementation of Learning Software of Wavelet Image Transform

Kiyomitsu KAJISA

This technical paper reports implementation of learning software of wavelet image transform using a C# language. First, abstracts of the main functions of this learning software, i.e. wavelet transform, wavelet compression, image saving, evaluation of image quality, and information embedding/detection, are reported. Then, some experimental results using this learning software are shown, including processing time. This learning software is supposed to be not only useful for students, who first learn wavelet transform, but also useful for experimenting information embedding in images using wavelet transform. It should be noted that the zerotree method for compressing the wavelet coefficients is not included in this learning software.

Keywords : Wavelet Transform, Compression, Information Embedding, C Sharp

1 まえがき

ウェーブレット画像変換の学習ソフトを試作したので、その機能の概要と学習ソフトを用いた実験例について報告する。

ウェーブレット画像変換は画像をウェーブレットにより周波数変換する変換方式で、画像圧縮の国際標準 JPEG 2000¹⁾の基本方式としても採択された方式である (JPEG2000 は JPEG に比べてブロック間歪が少ない特徴がある)。

代表的なウェーブレット (wavelet, さざ波) に平均 (低周波数成分) と差分 (高周波数成分) を基本とする Haar (ハール) と、4, 6 あるいは 8 個の数値で表現する Daubechies (ドベシィ) がある。ウェーブレット変換の詳細については文献 1)~4) を参照されたい。2 次元のウェーブレット画像変換は、一つの画像に対し、Haar などのウェーブレットを使用し、水平方向の 1 次元ウェーブレット変換を垂直方向に順次施した後に、垂直方向の 1 次元ウェーブレット変換を水平方向に順次施すことを、変換領域が 1 画素になるまで、 $1/2^n$ ごとに繰り返す (図 1 を参照)。

言い換えると、一つの画像は、まず、四つのサブバンド (低周波数成分 LL1 と三つの高周波数成分 LH1, HL1, HH1) に再構成され、次に、LL1 はさらに四つのサブバンドへ再構成され、LLn が 1 画素になるまで

この処理が繰り返される。このようにして、変換されたウェーブレット係数が得られる。

本学習ソフトはウェーブレット画像変換を視覚的に学習するソフトであるが、加えて、ウェーブレット変換を用いた情報の埋め込み (電子透かし)^{5), 6)} も学習できるソフトである。

本学習ソフトの設計指針を次に掲げる。

- (1) メニュー方式による機能や設定の選択ではなく、何を選択したかが視覚的にわかるように、ラジオボタン、チェックボタン、機能選択ボタンを常にウィンドウ内に表示する。
- (2) 画像処理に関する基本的な処理結果はラベル表示する。機能ごとの注意事項, エラー, 処理結果はメッセージボックスで使用者へ確認を求める。
- (3) 原画像や処理結果の画像を別ウィンドウで表示せ

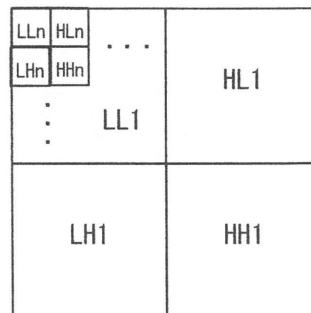


図 1 ウェーブレット変換

[†]情報工学科

ず、比較が容易なように固定した二つの画像領域のみを用いて画像表示する。

- (4) 処理後の画像を記録や再利用ができるように画像保存の機能を設ける。

なお、本学習ソフトの試作には、MS Visual Studio. NET⁷⁾の C# (シー・シャープ) を用いた。C#は Visual C++と Visual Basic のオブジェクト指向と視覚化の利点を取り入れた新たな言語である。また、本学習ソフトのウェーブレット変換とウェーブレット圧縮の機能は文献 4)を参考にしてプログラム化した。

2 学習ソフトの概要

本学習ソフトの主な機能は、ウェーブレット変換、ウェーブレット圧縮、画像保存、画質評価、および情報埋め込み・検出である。それぞれ試作した学習ソフトの実行ボタンに対応している。以下、それぞれの機能の概要について述べる。参考までに、本学習ソフトのファイル構成を図 2 に示す。

2.1 ウェーブレット変換

本学習ソフトのウェーブレット変換(Wavelet Transform)は、Haar あるいは Daubechies ウェーブレットを用いて、BMP 画像をウェーブレット変換し、視覚的にわかりやすいように、その結果であるウェーブレット変換係数を画像として出力する機能である。

図 3 の上段に、指定したグレースケールの BMP 画

像を読み込み、ウェーブレット変換を実施した直後の状態を示す。ラベルには原画像とウェーブレット変換係数に関する情報が表示される。もし、8 ビット/画素の原画像がカラーマップ画像の場合は、カラーマップに従い、グレースケール画像へ変換される。

次に、図 3 下段の右側に示すように、ウェーブレット変換係数を画素の濃度値に変換して画像表示する。ここでは、係数を 20 倍し、128 を加え、0~255 の範囲に量子化しているので、より正の係数はより明るく、より負の係数はより暗く、0 に近い係数は中間色で表示される。図 3 下段右側の画像より、低周波数成分の領域では係数の変化が多く、高周波数成分の領域では輪郭部で大きい正負の係数が出現する様子が観察される。

さらに、ウェーブレット変換係数は、設定された変換しきい値[%]のより大きい非ゼロ係数を保持するように、その他のより小さい変換係数はゼロへ強制的に変換される。図 3 下段の左側に、残った非ゼロ係数を黒画素として表示した画像を示す。図 3 下段のラベル表示より、変換しきい値が 50[%]の場合、最小の係数(絶対値)は 3 であることがわかる。この値は、変換係数の絶対値をソートで並べ替えて得た結果で、ゼロへの変換に使用される基準値である。

このようにして削減(decimate)された変換係数により画像圧縮が可能となり、これらの非ゼロ変換係数を用いてロシーな(lossy, 有歪の、非可逆の)復元画像が生成されることになる。

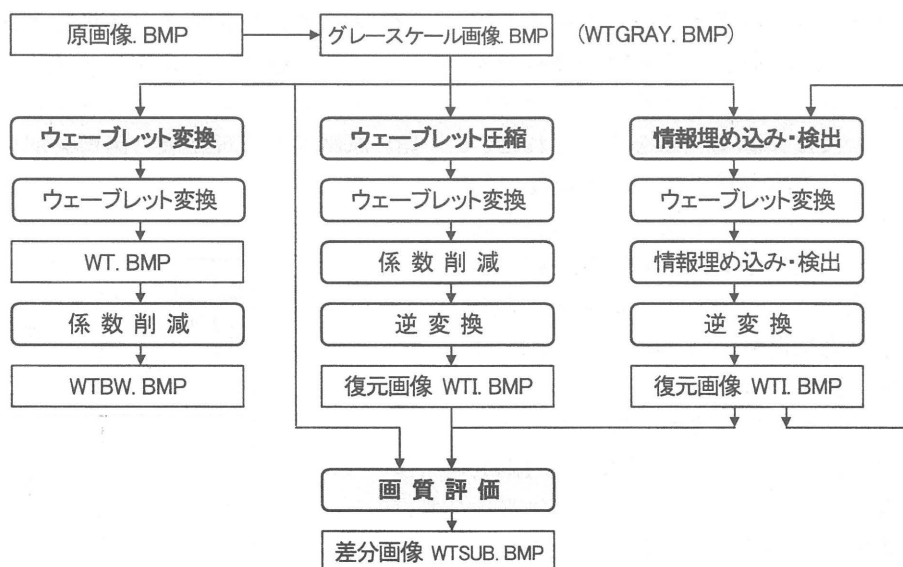


図2 ウェーブレット画像変換の学習ソフトのファイル構成

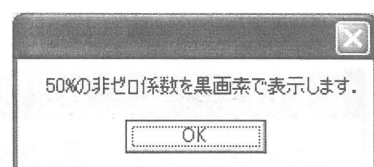
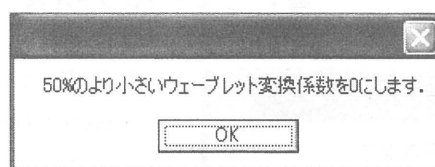
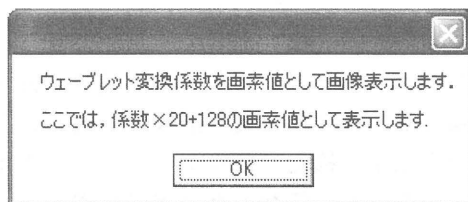


図3 ウェーブレット変換



図4 ウェーブレット圧縮

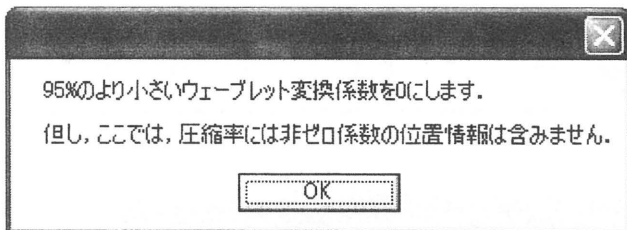


図5 ウェーブレット圧縮

2.2 ウェーブレット圧縮

ウェーブレット圧縮 (Wavelet Compression) は、Haar あるいは Daubechies ウェーブレットを用いて、BMP 画像をウェーブレット画像圧縮する機能である。圧縮には前述の非ゼロ係数削減(decimation)の手法を用いている。JPEG2000 で使用されているウェーブレットゼロツリー符号化(wavelet zerotree encoding)の手法は用いていない。また、この機能には圧縮ファイルの生成は含まれていない (図5を参照)。

非ゼロ係数削減は、前述したように、ウェーブレット変換係数の大きい $x\%$ (圧縮率で設定) は保持し、残りの $(100 - x)\%$ の変換係数はすべて 0 に変換する手法である。図4の圧縮率 5% の場合、同図中のラベル表示に示すように、ゼロ係数は 3% から 95% となるように、48.375 より小さいウェーブレット係数はゼロに変更される。このようにして変更されたウェーブレッ

ト係数に対してウェーブレット逆変換 (Inverse Wavelet Transform) が施され、復元画像が生成される。

ウェーブレット圧縮の機能は、ウェーブレット変換の機能からウェーブレット係数の画像表示を省き、逆変換による復元画像の表示を加えたものである。図4右側に圧縮率 5% でウェーブレット圧縮したときの復元画像を示す。原画像と対比し、画質劣化の様子がよくわかる。

2.3 画像保存

本学習ソフト内部で生成される BMP 中間ファイルは、図2中に示したように、WT. BMP, WTBW. BMP および WTI. BMP である。WT. BMP は変換しきい値を用いたウェーブレット変換により、WTBW. BMP は非ゼロ係数削減により生成され、画面表示される。WTI. BMP は、ウェーブレット圧縮の最後のウェーブレット逆変換により復元画像として生成され、画面表示される。

これらの BMP ファイルはそれぞれ対応する保存ボタンを用い、名前を付けて保存できる。名前を付けて保存するときのデフォルトのファイル名は、例えば WTI. BMP の場合、XXxxWTI. BMP である。ここで、XX は原画像のファイル名、xx は指定された圧縮率を示す。このデフォルトのファイル名 (例えば、LENA10 WTI. BMP) は実験データの記録保存には有用である。

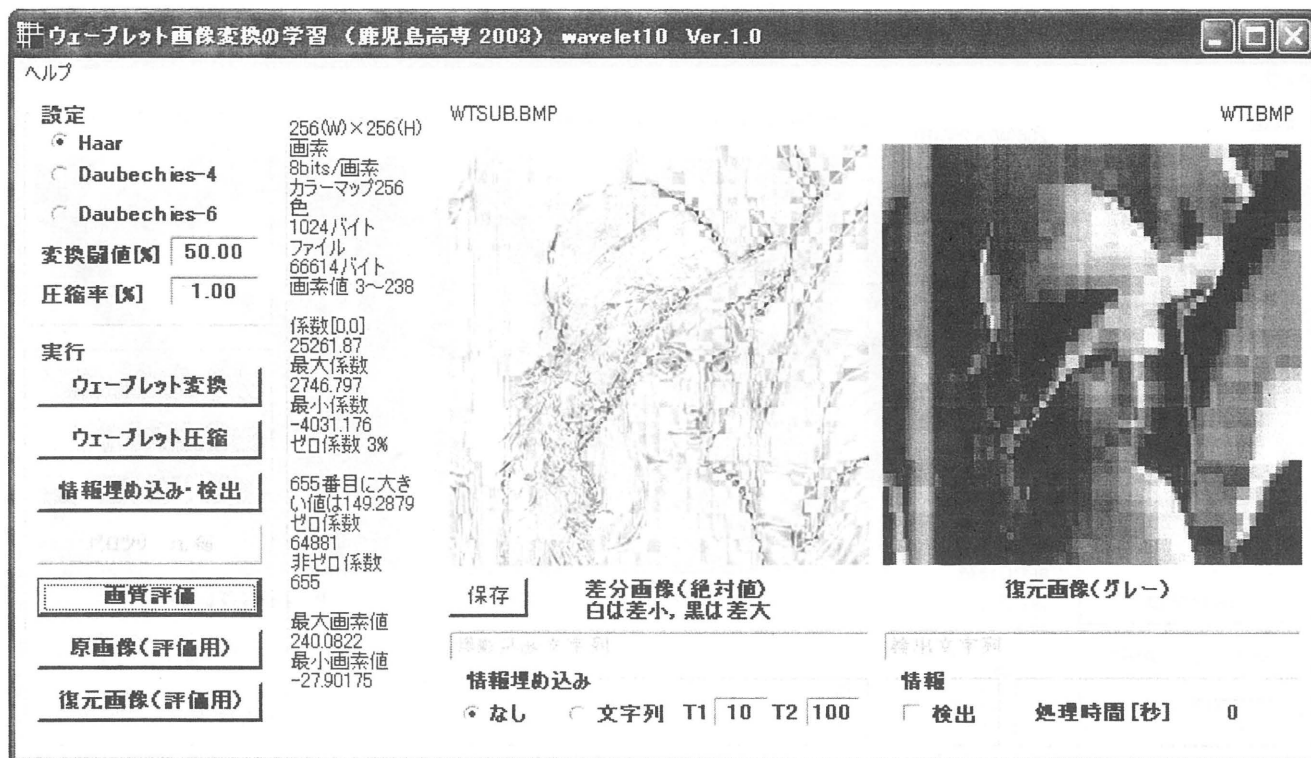


図6 画質評価

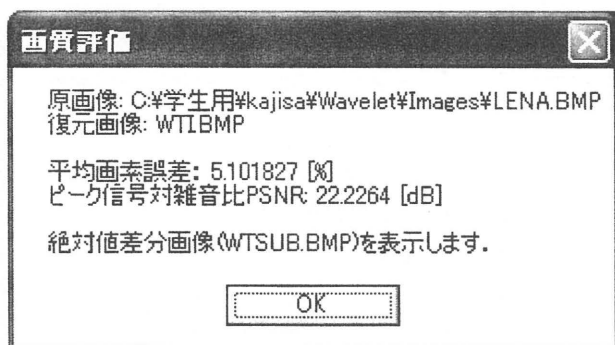


図7 画質評価

2.4 画質評価

画質評価は、ウェーブレット圧縮あるいは情報埋め込みにより生成された復元画像の原画像に対する画質劣化を評価する機能である。保存された原画像と復元画像の BMP ファイルをロードすることによっても評価できる。両画像の絶対値差分は差分画像(WTSUB.BMP)として表示される。図6の右側に復元画像、左側に差分画像を示す。大きい差分は輪郭部に出現する。差分画像は濃度値の低い、薄い部分が鮮明に印刷されないが、画面表示ではその差を観察することができる。

画質評価の結果表示を図7に示す。画質劣化は、平均画素誤差(Average Pixel Error)とピーク信号対雑音比(PSNR: Peak Signal-to-Noise Ratio)により表現す

る。平均画素誤差は次式により算出される。

$$(\sum \sum |f_{ij} - d_{ij}|) / MN / 255, i = 1..M, j = 1..N \quad (1)$$

ここで、 f は原画像の画素値、 d は復元画像の画素値、 M, N は縦横の画素数である。平均画素誤差は画素当たりの平均誤差の 255 レベルに対する割合である。

また、ピーク信号対雑音比は次式により算出される。

$$RMS = \sqrt{(\sum \sum |f_{ij} - d_{ij}|^2) / MN} \quad (2)$$

$$PSNR = 20 \log_{10} (255 / RMS) \quad (3)$$

ここで、 $\sqrt{}$ は平方根であり、 RMS は平均二乗誤差 (root mean square error)を示す。

2.5 情報の埋め込み

以上がウェーブレット変換と圧縮に関する基本的な学習内容であるが、この学習ソフトには簡単な情報埋め込み(embed)・検出(detect)の機能が付いている。ここでの情報埋め込みは、使用者が文字列を入力し、その入力された文字列の 1/0 のビット列 (図9中央の埋め込み情報を参照) を基に指定範囲のウェーブレット係数がある値に置き換えることにより実現される。図8の右側に文字列「Hayato」を埋め込んだ復元画像を示すが、この例での平均画素誤差は 0.205%、PSNR は 48.5dB と、原画像との肉眼による区別は難しい。また、この例では、図11(a)の最後の行に示すように埋め込み可能な最大ビット数は 8159 ビットである。



図8 情報埋め込み

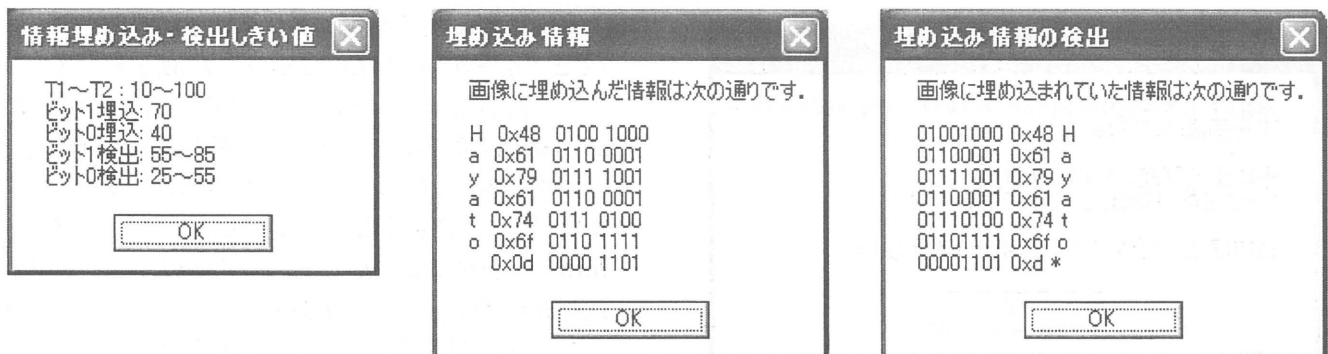


図9 情報埋め込みと検出

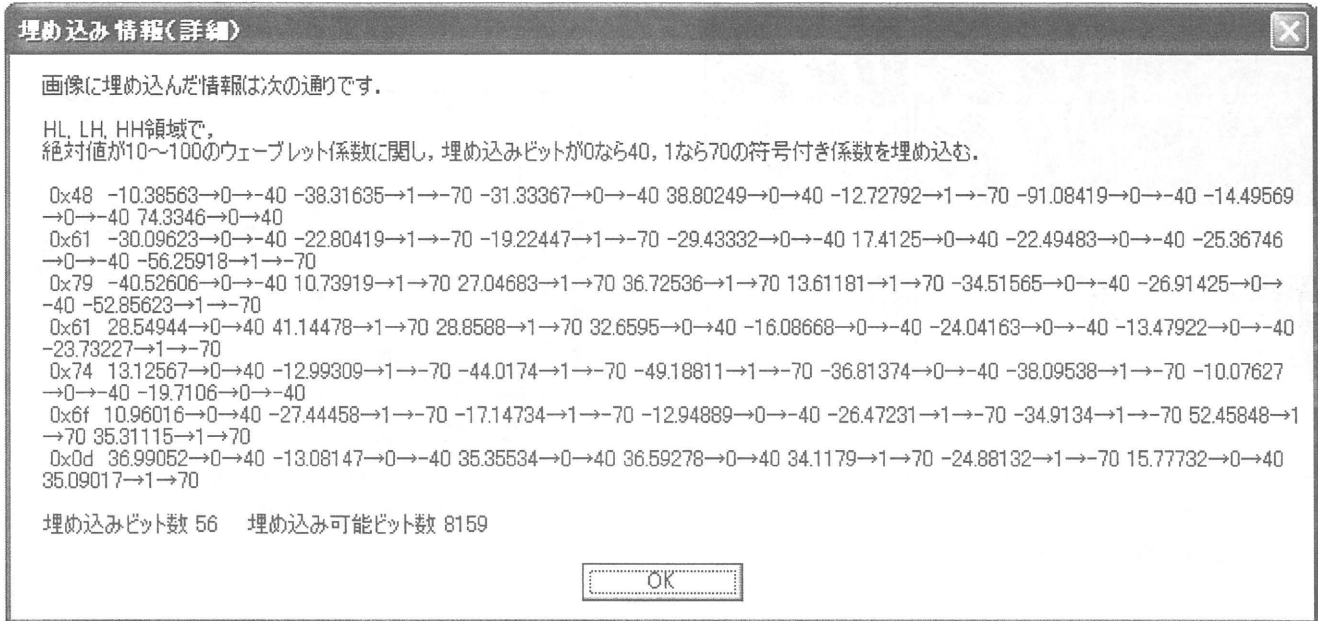
情報の埋め込みにあたっては、係数を置き換える指定範囲($T1 \sim T2$, 初期値は $10 \sim 100$)を変更し、実験することもできる。ここでは、ビット 0, 1 の埋め込み値 $C0, C1$, ビット 0 の検出範囲 $C0L \sim C0H$, およびビット 1 の検出範囲 $C1L \sim C1H$ は、正の場合、次式で与えられる。

$$\left. \begin{aligned} C0 &= (T2-T1) \cdot 2/6 + T1 \\ C1 &= (T2-T1) \cdot 4/6 + T1 \\ C0L &= (T2-T1) \cdot 1/6 + T1 \\ C0H &= (T2-T1) \cdot 3/6 + T1 \\ C1L &= (T2-T1) \cdot 3/6 + T1 \\ C1H &= (T2-T1) \cdot 5/6 + T1 \end{aligned} \right\} \quad (4)$$

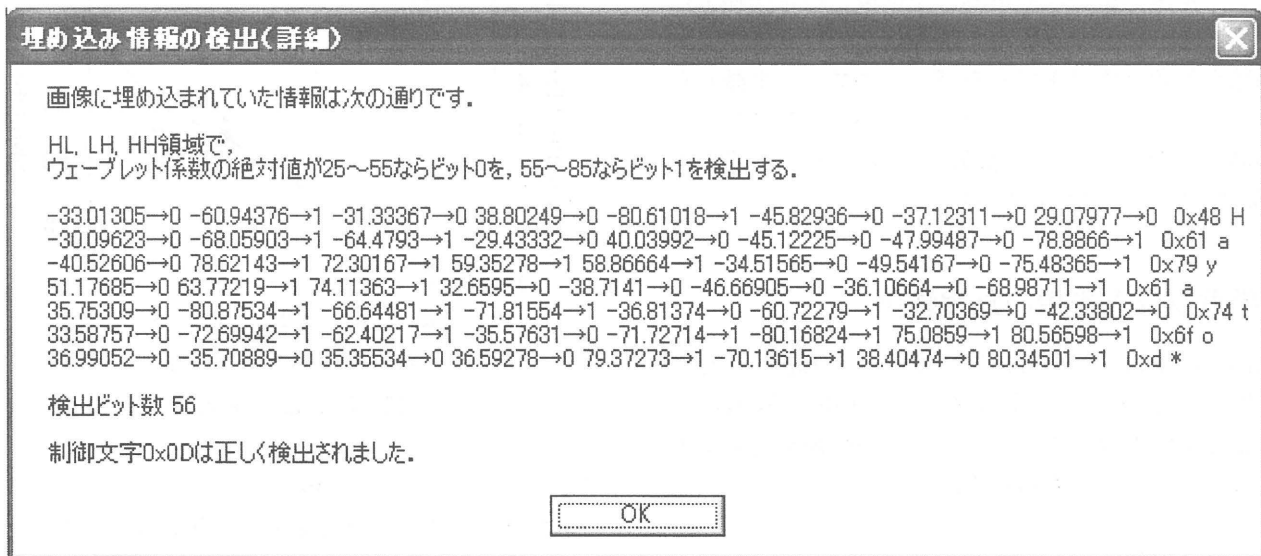


図10 情報埋め込み・検出のしきい値

しきい値については、図 10 のしきい値と図 9 左側の



(a) 埋め込み情報 (詳細)



(b) 埋め込み情報の検出 (詳細)

図 11 情報埋め込みと検出 (詳細)

しきい値の例を参照されたい。これは文献 8)の埋め込み値 $T1$, $T2$, 検出しきい値 $(T1+T2)/2$ を改良した独自の手法である。

ここでの情報の埋め込みは低周波数領域 LL1 を除く、高周波数領域に対して行う。ウェーブレット係数の置き換えの例を図 11 (a) に示す。

この手法によると、検出時に原画像を参照することなく、復元画像のみから埋め込み情報が検出できる。また、攻撃に対しても耐性があることが想定される。

2.6 埋め込み情報の検出

情報を埋め込んだウェーブレット係数はウェーブレット逆変換により情報を埋め込んだ画像として復元される。埋め込み情報の検出はこの復元画像を原画像としてウェーブレット変換を施し、高周波数領域のウェーブレット係数が式(4)で示した検出範囲内にあれば、ビット 0 あるいはビット 1 として検出される。

文字列「Hayato」を検出した例を図 9 の右側と図 11 (b) に示す。現バージョンでは、英数字に加えて 2 バイトコードの漢字も埋め込み、コードとしては検出

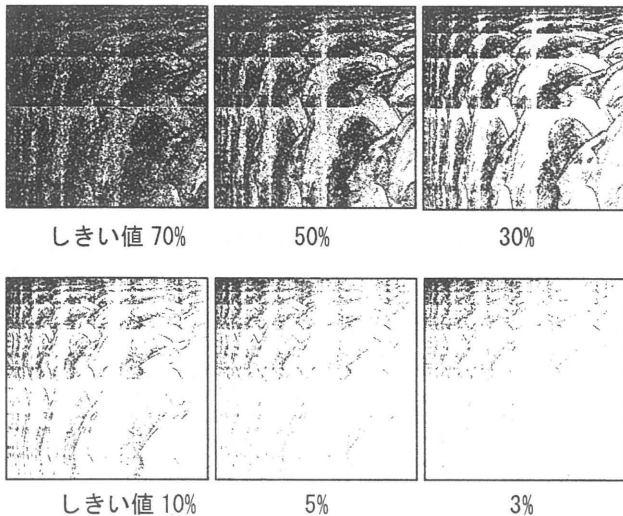


図 12 ウェーブレット変換 (Lena, Haar)

できるが、検出文字列の表示は未サポートである。

なお、本来、情報埋め込みは圧縮後（指定圧縮率に対応するように非ゼロ係数を削減後）のウェーブレット係数を対象とすべきであるが、ここでは、削減しないウェーブレット係数を対象としている。そのため、検出時は当然であるが、情報埋め込み時も圧縮率を 100% に設定しておく必要がある。

3 学習ソフトを用いた実験例

試作した学習ソフトを使用して行った実験の例を以下に掲げる。授業でこの学習ソフトを使用する場合は、学生ごとに原画像やウェーブレットの種類を変えて実験することなどが考えられる。

3.1 ウェーブレット変換

ウェーブレット変換後のウェーブレット係数に対し、変換しきい値を変化させて設定した場合の出力画像を図 12 に示す。変換しきい値の割合の係数が非ゼロ係数として保持され、黒画素で表現されている。1/20 圧縮に対応する 5% では、大半の高周波数成分の係数がゼロとなり、主に低周波数成分の係数を使用して画像が復元されることがわかる。

3.2 ウェーブレット圧縮

学習ソフトの画質評価の機能を用い、Haar ウェーブレット変換係数に対し、圧縮率を変化させた場合の平均画素誤差と PSNR の特性を図 13 に示す。圧縮率が 100% のときの平均画素誤差は 0% で、PSNR は 200dB である。

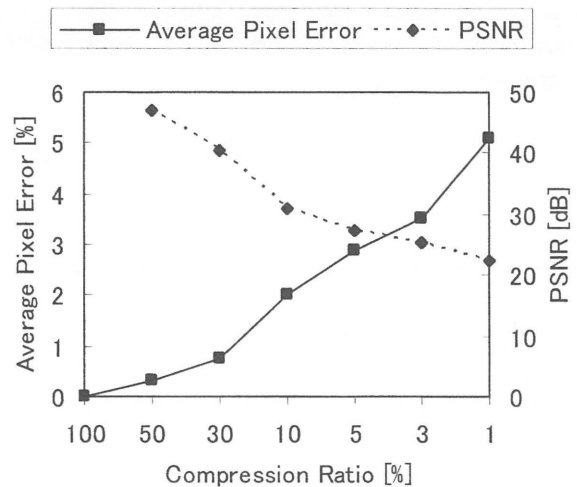


図 13 平均画素誤差と PSNR (Lena, Haar)

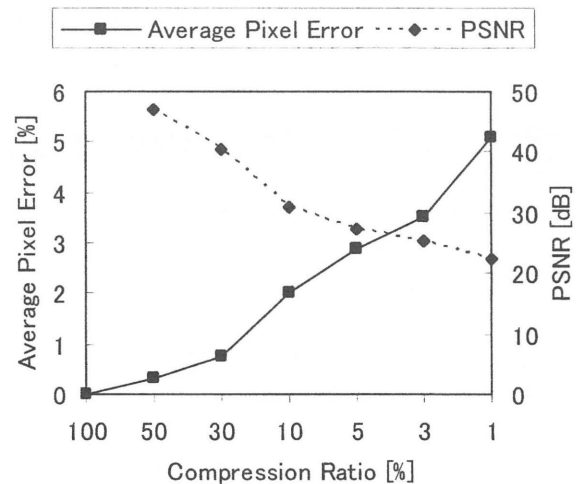


図 14 PSNR (Lena)

また、Haar, Daubechies-4, Daubechies-6 ウェーブレットを使用したときの PSNR の特性を図 14 に示す。この例では、Haar ウェーブレットに対し Daubechies では 1~2dB の改善がみられるが、Daubechies-4 と Daubechies-6 では大差はないことがわかる。

3.3 処理時間

2GHz の Pentium 4 を搭載したパソコンを使用した場合、原画像の読み込み・表示からウェーブレット変換係数を得るまでは約 0.3 秒、クイックソートで指定された圧縮率の非ゼロ係数を残し、ウェーブレット逆変換の後に復元画像を表示するまでは約 0.3 秒と高速に処理できる。これまでの実験例で使用した標準画像 Lena は 256×256 画素のグレースケール画像であるが、512×512 画素のグレースケール画像を用いると前半が約 2.3 秒、後半が約 2.3 秒の処理時間を要する。

表 1 情報埋め込みと画質評価 (Lena)

ウェーブレット	Haar	DB-4	DB-6
埋め込み可能ビット数 [bit]	8159	6119	5743
埋め込み文字数+0x0D [Byte]	1019	764	717
埋め込みビット数 [bit]	8152	6112	5736
平均画素誤差 [%]	3.158	2.600	2.551
PSNR [dB]	25.91	27.07	27.21

ただし、本学習ソフトの現バージョンで、次に述べる非常に長い文字列を埋め込む実験を行う場合は、図 9 と図 11 に示した埋め込み情報と埋め込み情報の検出のデータを表示するメッセージボックス内の、実際には表示されない、表示データの生成に非常に時間がかかる欠点がある。この点は今後、改良する必要がある。

3.4 情報の埋め込み

本学習ソフトは文字列情報を埋め込み、埋め込んだ情報を文字列として検出することはできるが、実験のために埋め込み可能ビット数の 2 値情報（擬似乱数あるいは 1/0 の交互データ）を埋め込む機能は現バージョンでは有していない。そこで、ここでは、1/0 の交互データより成る文字'U'(0x55)の連続した文字列をメモ帳で作成し、それを本学習ソフトの埋め込み文字列の入力欄にコピーすることにより、実験を進める。

ウェーブレットの種類を変えて、埋め込み可能ビット数に相当する文字数を埋め込んだ実験結果を表 1 に示す。使用した標準画像は今までの実験と同じく 256 × 256 画素の Lena である。Daubechies ウェーブレッ



図 15 1019 文字を埋め込んだ画像 (Lena, Haar, 25.9dB)



図 17 100 文字を埋め込んだ画像 (Lena, Haar, 36.3dB)

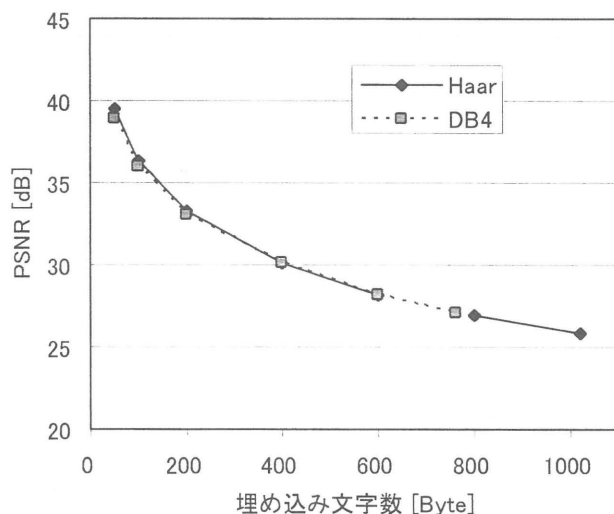


図 16 埋め込み文字数と PSNR (Lena)



図 18 原画像 (Lena, 256 × 256 画素)

トよりも Haar ウェーブレットを用いた方が、埋め込み可能ビット数は大きいことがわかる⁹⁾。図 15 に Haar ウェーブレットを用いた場合の復元画像を示すが、画質劣化が著しく、縦筋が目立つ。

次に、同様の方法で埋め込み文字数と PSNR との関係を実験した結果を図 16 に示す。Haar よりも Daubechies-4 ウェーブレットの方がわずかに良好な PSNR であるが、大差はない。35dB を視覚的に画質劣化の少ない画像の境界値とみなす¹⁰⁾と、いずれのウェーブレットの場合も百数十文字が実用上の埋め込み可能な文字数とみなすことができる。図 17 に Haar ウェーブレットを用いて 100 文字を埋め込んだ画像を示す。また、画質の比較のために、図 18 に Lena の原画像を示す。

4 むすび

本報告では、試作したウェーブレット画像変換の学習ソフトの機能概要と、その学習ソフトを用いた実験例について報告した。画像処理の分野の学習においては、肉眼で画像を観察することは非常に重要であるため、本学習ソフトはウェーブレット変換の原理と効果を視覚的に体得するために有用であると思われる。また、本学習ソフトのプログラムを基に、改良することにより、各種の情報埋め込み手法を研究することも可能である。

なお、本学習ソフト（初版）は削減した非ゼロ係数を符号化するウェーブレットゼロツリー符号化の機能はサポートしていない。今後の更新が望まれる。

謝辞

本研究は平成 14 年度校内研究助成金（教育改善プロジェクト）を受けて行われたことを記して謝意を表します。

参考文献

- 1) D.S. Taubman and M.W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publications, 2002.
- 2) 中野宏毅, 山本鎮男, 吉田靖夫, ウェーブレットによる信号処理と画像処理, 共立出版, 1999.
- 3) 鎌田清一郎, 画像処理, サイエンス社, 2003.
- 4) S. Welstead, Fractal and Wavelet Image Compression Techniques, SPIE Optical Engineering Press, 1999.
- 5) 松田甲子雄, 電子透かしの基礎, 共立出版, 1998.
- 6) 小野東, 電子透かしとコンテンツ保護, オーム社, 2001.
- 7) Microsoft Visual Studio.NET Professional Edition, Version 2002.
- 8) H. Inoue, A. Miyazaki, A. Yamamoto and T. Katsura, "A digital watermark technique based on the wavelet transform and its robustness on image compression and transformation," IEICE Trans. Fundamentals, Vol.E82-A, no.1, pp.2-10, Jan. 1999.
- 9) 竹山征宏, 加治佐清光, "ウェーブレット変換を用いた画像への情報埋め込みに関する考察," 第 11 回電子情報通信学会九州支部学生会講演会講演論文集, p.12, Sep. 2003.
- 10) H.M. Wang, P.S. Su and C.J. Kuo, "Wavelet-based digital image watermarking," Optics Express, vol.3, no.12, pp.491-496, Dec. 1998.